**Biochemical analysis of photosynthetic metabolites**

#Biochemical analysis of photosynthetic metabolites

setwd("/Users/alish/OneDrive/Documents/Uni/MRes/Raw\_et\_al")

#Install.packages

library(ggplot2)

library(readr)

library(Cairo)

# Load Data

data <- read\_csv("cleaned\_metabolite\_data.csv")

# Ensure Treatment is a factor with correct ordering

data$Treatment <- factor(data$Treatment, levels = c("Control", "Fumigated"))

# Remove missing or infinite values

data <- na.omit(data)

data$Treatment <- factor(data$Treatment, levels = c("Control", "Fumigated"))

# Remove missing or infinite values

data <- na.omit(data)

# Add units for each metabolite

metabolite\_units <- c(

"Glutathione (nmol/g^-1 DW)", "Cysteine (nmol g^-1 DW)",

"Neoxanthin (nmol mg^-1 DW)", "Violaxanthin (nmol mg^-1 DW", "Lutein (nmol mg^-1 DW)",

"Chlorophyll a (nmol mg^-1 DW)", "Chlorophyll b (nmol mg^-1 DW)",

"α-Carotene (nmol mg^-1 DW)", "β-Carotene (nmol mg^-1 DW)",

"δ-Tocopherol (nmol mg^-1 DW)", "α-Tocopherol (nmol mg^-1 DW)" ,"Glu\_Cys"

)

# Rename metabolites with units

data$Metabolite <- factor(data$Metabolite, levels = unique(data$Metabolite), labels = metabolite\_units)

# Conduct t-tests for each metabolite

results <- data.frame(Metabolite = character(),

p\_value = numeric(),

stringsAsFactors = FALSE)

for (metabolite in unique(data$Metabolite)) {

# Subset the data for the current metabolite

subset\_data <- data[data$Metabolite == metabolite, ]

# Perform t-test for Normalized Value between Control and Fumigated

t\_test\_result <- t.test(subset\_data$Normalised.Value ~ subset\_data$Treatment)

# Store the results

results <- rbind(results, data.frame(Metabolite = metabolite, p\_value = t\_test\_result$p.value))

}

# Adjust for multiple comparisons using Benjamini-Hochberg (False Discovery Rate)

results$p\_adj <- p.adjust(results$p\_value, method = "BH")

results <- data.frame(Metabolite = character(),

t\_test\_p\_value = numeric(),

wilcox\_p\_value = numeric(),

t\_test\_p\_adj = numeric(),

wilcox\_p\_adj = numeric(),

stringsAsFactors = FALSE)

# Perform t-test and Wilcoxon test for each metabolite

for (metabolite in unique(data$Metabolite)) {

# Subset the data for the current metabolite

subset\_data <- data[data$Metabolite == metabolite, ]

# Perform t-test for Normalized Value between Control and Fumigated

t\_test\_result <- t.test(subset\_data$Normalised.Value ~ subset\_data$Treatment)

# Perform Wilcoxon test for Normalised Value between Control and Fumigated

wilcox\_test\_result <- wilcox.test(subset\_data$Normalised.Value ~ subset\_data$Treatment)

# Store the results

results <- rbind(results, data.frame(

Metabolite = metabolite,

t\_test\_p\_value = t\_test\_result$p.value,

wilcox\_p\_value = wilcox\_test\_result$p.value

))

}

# Adjust for multiple comparisons using Benjamini-Hochberg (False Discovery Rate)

results$t\_test\_p\_adj <- p.adjust(results$t\_test\_p\_value, method = "BH")

results$wilcox\_p\_adj <- p.adjust(results$wilcox\_p\_value, method = "BH")

# Print the results

print(results)

write.csv(results, "statistical\_results.csv", row.names = FALSE)

# \*\*Add units for each metabolite\*\*

metabolite\_units <- c(

"Glutathione (nmol/g^-1 DW)", "Cysteine (nmol g^-1 DW)",

"Neoxanthin (nmol mg^-1 DW)", "Violaxanthin (nmol mg^-1 DW", "Lutein (nmol mg^-1 DW)",

"Chlorophyll a (nmol mg^-1 DW)", "Chlorophyll b (nmol mg^-1 DW)",

"α-Carotene (nmol mg^-1 DW)", "β-Carotene (nmol mg^-1 DW)",

"δ-Tocopherol (nmol mg^-1 DW)", "α-Tocopherol (nmol mg^-1 DW)"

)

# Rename metabolites with units

data$Metabolite <- factor(data$Metabolite, levels = unique(data$Metabolite), labels = metabolite\_units)

# Create boxplot

plot <- ggplot(data, aes(x=Treatment, y=Normalised.Value, fill=Treatment)) +

geom\_boxplot() +

scale\_fill\_manual(values=c('dodgerblue3', 'orange1')) +

geom\_jitter(alpha=0.3, position=position\_jitter(0.1)) +

theme(panel.background = element\_blank(),

axis.line = element\_line(color = 'black', linewidth = 0.3),

plot.title = element\_text(size = 12),

legend.position = 'none',

axis.text.y = element\_text(size=15),

axis.text.x = element\_blank(),

axis.title.x = element\_blank(),

axis.title.y = element\_text(size=15),

axis.ticks.x = element\_blank()) +

ylab("Normalised Metabolite Level") +

facet\_wrap(~Metabolite, scales="free\_y")

CairoPNG("metabolite\_boxplots\_august.png", width = 18, height = 15, units = "in", res = 300)

print(plot)

dev.off()